

Neural Machine Transliteration: Preliminary Results

Amir H. Jadidinejad

Bayan Inc.

jadidinejad@bayan.co.ir

Abstract

Machine transliteration is the process of automatically transforming the script of a word from a source language to a target language, while preserving pronunciation. Sequence to sequence learning has recently emerged as a new paradigm in supervised learning. In this paper a character-based encoder-decoder model has been proposed that consists of two Recurrent Neural Networks. The encoder is a Bidirectional recurrent neural network that encodes a sequence of symbols into a fixed-length vector representation, and the decoder generates the target sequence using an attention-based recurrent neural network. The encoder, the decoder and the attention mechanism are jointly trained to maximize the conditional probability of a target sequence given a source sequence. Our experiments on different datasets show that the proposed encoder-decoder model is able to achieve significantly higher transliteration quality over traditional statistical models.

1 Introduction

Machine Transliteration is defined as phonetic transformation of names across languages (Zhang et al., 2015; Karimi et al., 2011). Transliteration of named entities is the essential part of many multilingual applications, such as machine translation (Koehn, 2010) and cross-language information retrieval (Jadidinejad and Mahmoudi, 2010).

Recent studies pay a great attention to the task of Neural Machine Translation (Cho et al., 2014a; Sutskever et al., 2014). In neural machine translation, a single neural network is responsible for reading a source sentence and generates its trans-

lation. From a probabilistic perspective, translation is equivalent to finding a target sentence y that maximizes the conditional probability of y given a source sentence x , i.e., $\arg \max_y p(y | x)$. The whole neural network is *jointly* trained to maximize the conditional probability of a correct translation given a source sentence, using the bilingual corpus.

Transforming a name from spelling to phonetic and then use the constructed phonetic to generate the spelling on the target language is a very complex task (Oh et al., 2006; Finch et al., 2015). Based on successful studies on Neural Machine Translation (Cho et al., 2014a; Sutskever et al., 2014; Hirschberg and Manning, 2015), in this paper, we proposed a character-based encoder-decoder model which learn to transliterate end-to-end. In the opposite side of classical models which contains different components, the proposed model is trained end-to-end, so it able to apply to any language pairs without tuning for a specific one.

2 Proposed Model

Here, we describe briefly the underlying framework, called *RNN Encoder-Decoder*, proposed by (Cho et al., 2014b) and (Sutskever et al., 2014) upon which we build a machine transliteration model that learns to transliterate end-to-end.

The encoder is a character-based recurrent neural network that learns a highly nonlinear mapping from a spelling to the phonetic of the input sequence. This network reads the source name $x = (x_1, \dots, x_T)$ and encodes it into a sequence of hidden states $h = (h_1, \dots, h_T)$:

$$h_t = f(x_t, h_{t-1}) \quad (1)$$

Each hidden state h_i is a bidirectional recurrent representation with forward and backward sequence information around the i th character. The

representation of a forward sequence and a backward sequence of the input character sequence is estimated and concatenated to form a context set $C = \{h_1, h_2, \dots, h_T\}$ (Dong et al., 2015; Chung et al., 2016). Then, the decoder, another recurrent neural network, computes the conditional distribution over all possible transliteration based on this context set and generates the corresponding transliteration $y = (y_1, \dots, y_{T'})$ based on the encoded sequence of hidden states h .

The whole model is jointly trained to maximize the conditional log-probability of the correct transliteration given a source sequence with respect to the parameters θ of the model:

$$\theta^* = \arg \max_{\theta} \sum_{n=1}^N \sum_{t=1}^{T_n} \log p(y_t^n | y_{<t}^n, x^n), \quad (2)$$

where (x^n, y^n) is the n -th training pair of character sequences, and T_n is the length of the n -th target sequence (y^n). For each conditional term in Equation 2, the decoder updates its hidden state by:

$$h_{t'} = f(y_{t'-1}, h_{t'-1}, c_{t'}) \quad (3)$$

where $c_{t'}$ is a context vector computed by a soft attention mechanism:

$$c_{t'} = f_a(y_{t'-1}, h_{t'-1}, C) \quad (4)$$

The soft attention mechanism f_a weights each vector in the context set C according to its relevance given what has been transliterated.

Finally, the hidden state $h_{t'}$, together with the previous target symbol $y_{t'-1}$ and the context vector $c_{t'}$, is fed into a feedforward neural network to result in the conditional distribution described in Equation 2. The whole model, consisting of the encoder, decoder and soft attention mechanism, is trained end-to-end to minimize the negative log-likelihood using stochastic gradient descent.

3 Experiments

We conducted a set of experiments to show the effectiveness of RNN Encoder-Decoder model (Cho et al., 2014b; Sutskever et al., 2014) in the task of machine transliteration using standard benchmark datasets provided by NEWS 2015-16 shared task (Banchs et al., 2015). Table 1 shows different datasets in our experiments. Each dataset covers different levels of difficulty and training set size. The proposed model has been applied on

| TaskID | Source | Target | Data Size | | |
|--------|---------|---------|-----------|------|--------|
| | | | Train | Dev | Test |
| En-Ch | English | Chinese | 37K | 2.8K | 1.008K |
| Ch-En | Chinese | English | 28K | 2.7K | 1.019K |
| En-Th | English | Thai | 27K | 2.0K | 1.236K |
| Th-En | Thai | English | 25K | 2.0K | 1.236K |
| En-Hi | English | Hindi | 12K | 1.0K | 1.000K |
| En-Ta | English | Tamil | 10K | 1.0K | 1.000K |
| En-Ka | English | Kannada | 10K | 1.0K | 1.000K |
| En-Ba | English | Bangla | 13K | 1.0K | 1.000K |
| En-He | English | Hebrew | 9.5K | 1.0K | 1.100K |
| En-Pe | English | Persian | 10K | 2.0K | 1.042K |

Table 1: Datasets provided by NEWS 2015 (Banchs et al., 2015).

each dataset without tuning the algorithm for each specific language pairs. Also, we don't apply any preprocessing on the source or target language in order to evaluate the effectiveness of the proposed model in a fair situation. 'TaskID' is a unique identifier in the following experiments.

We leveraged a character-based encoder-decoder model (Bojanowski et al., 2015; Chung et al., 2016) with soft attention mechanism (Cho et al., 2014b). In this model, input sequences in both source and target languages have been represented as characters. Using characters instead of words leads to longer sequences, so Gated Recurrent Units (Cho et al., 2014a) have been used for the encoder network to model long term dependencies. The encoder has 128 hidden units for each direction (forward and backward), and the decoder has 128 hidden units with soft attention mechanism (Cho et al., 2014b). We train the model using stochastic gradient descent with Adam (Kingma and Ba, 2014). Each update is computed using a minibatch of 128 sequence pairs. The norm of the gradient is clipped with a threshold 1 (Pascanu et al., 2013). Also, beamsearch has been used to approximately find the most likely transliteration given a source sequence (Koehn, 2010).

Table 2 shows the effectiveness of the proposed model on different datasets using standard measures (Banchs et al., 2015). The proposed neural machine transliteration model has been compared to the baseline method provided by NEWS 2016 organizers (Banchs et al., 2015). Baseline results are based on a machine translation implementation at the character level using MOSES (Koehn et al., 2007). Experimental results shows that the proposed model is significantly better than the robust baseline using different metrics.

Figure 1 shows the learning curve of the pro-

| TaskID | Baseline | | | | Neural Machine Transliteration | | | |
|--------|----------|---------|--------|--------|--------------------------------|---------|--------|--------|
| | ACC | F-Score | MRR | MAP | ACC | F-Score | MRR | MAP |
| En-Ch | 0.1935 | 0.5851 | 0.1935 | 0.1830 | 0.2659 | 0.6227 | 0.3185 | 0.2549 |
| Ch-En | 0.0981 | 0.6459 | 0.0981 | 0.0953 | 0.0834 | 0.6564 | 0.1425 | 0.0830 |
| En-Th | 0.0680 | 0.7070 | 0.0680 | 0.0680 | 0.1456 | 0.7514 | 0.2181 | 0.1456 |
| Th-En | 0.0914 | 0.7397 | 0.0914 | 0.0914 | 0.1286 | 0.7624 | 0.1966 | 0.1286 |
| En-Hi | 0.2700 | 0.7992 | 0.2700 | 0.2624 | 0.3480 | 0.8349 | 0.4745 | 0.3434 |
| En-Ta | 0.2580 | 0.8117 | 0.2580 | 0.2573 | 0.3240 | 0.8369 | 0.4461 | 0.3235 |
| En-Ka | 0.1960 | 0.7833 | 0.1960 | 0.1955 | 0.2860 | 0.8224 | 0.4019 | 0.2856 |
| En-Ba | 0.2870 | 0.8360 | 0.2870 | 0.2837 | 0.3460 | 0.8600 | 0.4737 | 0.3438 |
| En-He | 0.1091 | 0.7715 | 0.1091 | 0.1077 | 0.1591 | 0.7976 | 0.2377 | 0.1582 |
| En-Pe | 0.4818 | 0.9060 | 0.4818 | 0.4482 | 0.5816 | 0.9267 | 0.7116 | 0.5673 |

Table 2: The effectiveness of neural machine transliteration is compared with the robust baseline (Koehn et al., 2007) provided by NEWS 2016 shared task on transliteration of named entities.

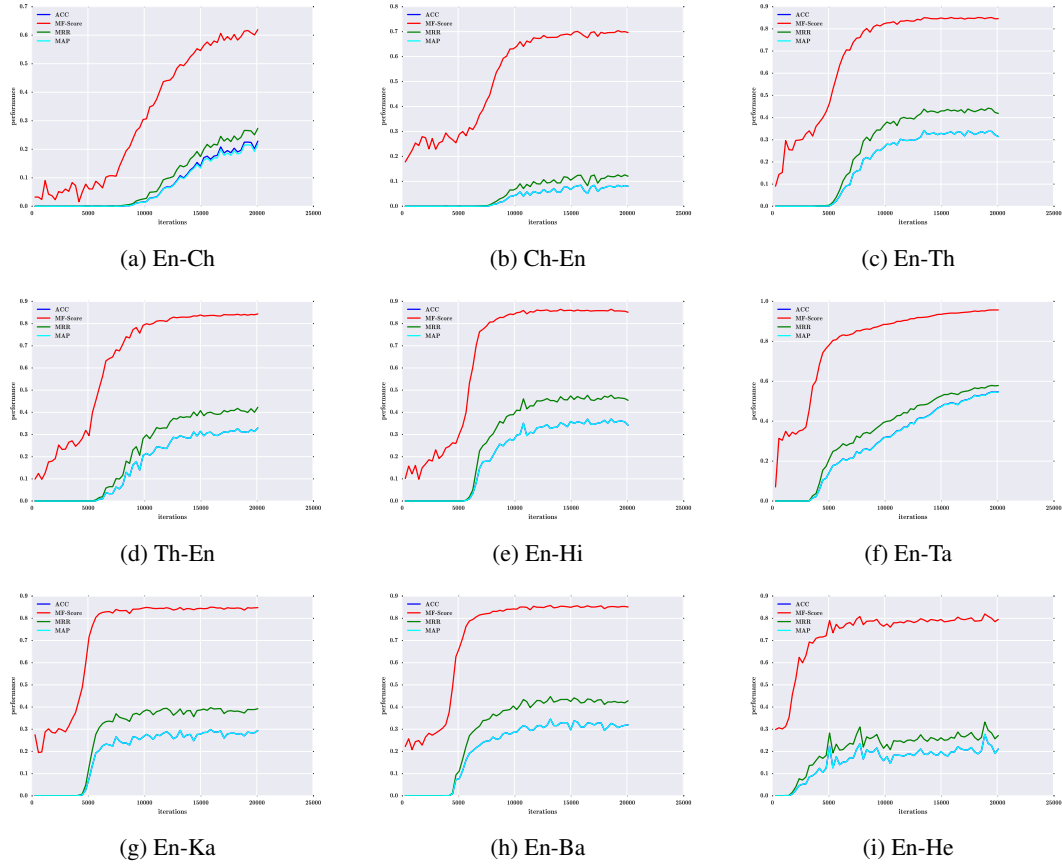


Figure 1: Learning curve of the proposed model on different datasets using the validation set. In most cases, the difference between 'ACC' and 'MAP' is negligible.

posed model on different datasets. It is clear that in most datasets, the trained model is capable of robust transliteration after a few number of iterations. As shown in Table 1, each dataset has different number of training set and also different number of characters in the source and target language. For example, when transliterating from English to Chinese (TaskID=‘En-Ch’) and English to Hebrew, the target names contains 548 and 37 different tokens respectively. Since we leverage a same model for different datasets without tuning the model for each dataset, differences in the learning curves are expectable. For some datasets (such as ‘En-Ch’), it takes more time to fit the model to the training data while for some others (such as ‘En-He’), the model fit to the training data after a few iterations.

4 Conclusion

In this paper we proposed Neural Machine Transliteration based on successful studies in sequence to sequence learning (Sutskever et al., 2014) and Neural Machine Translation (Ling et al., 2015; Costa-Jussà and Fonollosa, 2016; Bahdanau et al., 2015; Cho et al., 2014a). Neural Machine Transliteration typically consists of two components, the first of which encodes a source name sequence x and the second decodes to a target name sequence y . Different parts of the proposed model jointly trained using stochastic gradient descent to minimize the log-likelihood. Experiments on different datasets using benchmark measures revealed that the proposed model is able to achieve significantly higher transliteration quality over traditional statistical models (Koehn, 2010). In this paper we did not concentrate on improving the model for achieving state-of-the-art results, so applying hyperparameter optimization (Bergstra and Bengio, 2012), multi-task sequence to sequence learning (Luong et al., 2015) and multi-way transliteration (Firat et al., 2016; Dong et al., 2015) are quite promising for future works.

Acknowledgments

The authors would like to thank the developers of Theano (Theano Development Team, 2016) and DL4MT¹ projects. Also, the author would like to acknowledge the support of Bayan Inc. for research funding and computing support. The author

also thank Yasser Sourì for valuable comments.

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, volume abs/1409.0473.
- Rafael E. Banchs, Min Zhang, Xiangyu Duan, Haizhou Li, and A. Kumaran. 2015. Report of news 2015 machine transliteration shared task. In *Proceedings of the Fifth Named Entity Workshop*, pages 10–23, Beijing, China, July. Association for Computational Linguistics.
- James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(1):281–305, February.
- Piotr Bojanowski, Armand Joulin, and Tomas Mikolov. 2015. Alternative structures for character-level rnns. *arXiv preprint arXiv:1511.06303*.
- Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October. Association for Computational Linguistics.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.
- Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. *CoRR*, abs/1603.06147.
- Marta R. Costa-Jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. *CoRR*, abs/1603.00810.
- Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China, July. Association for Computational Linguistics.
- Andrew Finch, Lemao Liu, Xiaolin Wang, and Eiichiro Sumita. 2015. Neural network transduction models

¹<https://github.com/nyu-dl/dl4mt-tutorial>

- in transliteration generation. In *Proceedings of the Fifth Named Entity Workshop*, pages 61–66, Beijing, China, July. Association for Computational Linguistics.
- O. Firat, K. Cho, and Y. Bengio. 2016. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. *ArXiv e-prints*, January.
- Julia Hirschberg and Christopher D. Manning. 2015. Advances in natural language processing. *Science*, 349(6245):261–266.
- AmirHossein Jadidinejad and Fariborz Mahmoudi. 2010. Cross-language information retrieval using meta-language index construction and structural queries. In Carol Peters, GiorgioMaria Di Nunzio, Mikko Kurimo, Thomas Mandl, Djamel Mostefa, Anselmo Peñas, and Giovanna Roda, editors, *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, volume 6241 of *Lecture Notes in Computer Science*, pages 70–77. Springer Berlin Heidelberg.
- Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. 2011. Machine transliteration survey. *ACM Comput. Surv.*, 43(3):17:1–17:46, April.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.
- Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W. Black. 2015. Character-based neural machine translation. *CoRR*, abs/1511.04586.
- Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *CoRR*, abs/1511.06114.
- Jong-Hoon Oh, Key-Sun Choi, and Hitoshi Isahara. 2006. A comparison of different machine transliteration models. *J. Artif. Intell. Res. (JAIR)*, 27:119–151.
- Razvan Pascanu, Çağlar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. 2013. How to construct deep recurrent neural networks. *CoRR*, abs/1312.6026.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.
- Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May.
- Min Zhang, Haizhou Li, Rafael E. Banchs, and A. Kumar. 2015. Whitepaper of news 2015 shared task on machine transliteration. In *Proceedings of the Fifth Named Entity Workshop*, pages 1–9, Beijing, China, July. Association for Computational Linguistics.